REMARKS

Claims 1-11 are currently pending in the instant application. Claims 1, 5, 8, and 9 are rejected under 35 U.S.C. §102(e) as being anticipated by U.S. Patent No. 5,933,601 (Fanshier el al.)(hereinafter, "Fanshier"). Claims 2-4, 6, 7, 10 and 11 are rejected under U.S.C. §103(a) as being unpatentable over Fanshier in view of U.S. Patent No. 6,275,867 (Bendert et al.)(hereinafter, "Bendert"). Applicant respectfully traverses for the reasons set forth below and in the previous Response Including Amendment.

As per Applicant's Response Including Amendment dated October 23, 2002, Applicant's claimed invention seeks to provide a solution where a distributed application that requires centralized administration via a master node is running in a clustered computing environment, where administrative control resides in each node of the environment.

More specifically, as set forth in detail in the subject specification, it is appreciated that distributed network applications, such as that which might be written for BEA Systems Inc.'s Tuxedo® transaction manager and messaging middleware, often have defined the concept of a master machine that performs administration for the entire distributed application. (*Application No. 09/127,167*, p. 2) In the exemplary case of the Tuxedo environment, Logical Machines representing server machines are grouped together to define a domain. One of these Logical Machines is designated as the master, on which is running a DBBL process which performs administration for the entire Domain, including bringing a component online, taking a component offline, or checking the status of an individual component. (*Id.* at p.3).

A problem arises with such a distributed application when it is required to run in a clustered computing environment, such as by way of example, Microsoft Cluster Server (MSCS) in Microsoft® Windows NT®, Enterprise Edition, where administration is implemented on each of the connected systems (nodes) composing the cluster. As set forth in more detail in the specification, in the exemplary clustered environment, MSCS is controlled by the Cluster Service, which runs on each note of the cluster. (*Id.*). The Cluster Service spawns one or more Resource Monitors, each of which calls entry points in a Resource DLL, the latter of which implements the actions needed to bring the resource on-line or to take the resources off-line. (*Id.*). Thus, contrary to distributed network application that requires centralized administration via a master node, each node in the clustered computing environment maintains the administrative control.

Applicant's invention provides a solution to this problem. As set forth in claims 1, 4, 8, and 9, an administrative request from the clustered computing environment is first received at an originating node, and it is then determined whether the originating node is a designated master node for the distributed network application. If it is determined that the originating mode is not the designated master node, then the administrative request is routed to that mode which is the designated master node. (*See, Serial No. 09/127,167, claims 1, 4, 8 and 9*). In a preferred embodiment, this is accomplished an instance of a named pipe is created between the originating node and the designated master node, and passing the administrative request from the originating node to the designated master node via the named pipe. (*See, Serial No. 09/127,167; e.g., claims 2, 6, and 10*).

Fanshier discloses an object-based systems management methodology for distributed computer networks, such as (then) NCR's TOP END^{TM} system, which allows customers to create their own systems administration utilities. (*See Fanshier generally; also, Background of the Invention; Col. 3, lines 45-53)*. In relevant part, the system 10 is comprised of one or more nodes 12 interconnected by a network 14, wherein each of the nodes 12 comprises one or more computers. (*Id., Col. 2, lines 31-33*). Having a client/server architecture, the system 10 includes server systems operating on node 12, and connections from at least one of the nodes 12 to the workstations 18 on which are operating client systems. (*Id. at Col 2, lines 37-43*). A node includes several modular components 20, each modular component 20 comprising a process or logical group that performs one of more functions, and which works with the other modular components 20 to process distributed transactions initiated by a client system. Work is divided among the nodes 12 by spreading the location of the modular components 20 across the nodes, and having one or more of these "spread-out" components execute a portion of the client-initiated request. (*Id. at Col 2, lines 50-64*). These modular, distributed application components 30 might include software such as an Oracle® accounting application, Sybase® airline application, and similar on-line transaction processing (OLTP) applications (*See, e.g., Id. at Fig. 1*).

Each node 12 of the system 10 further includes a node manager 28, which coordinates processing among all of the nodes 12. (*Id. at Col 3, lines 9-12; Figure 1*). The coordinating processes provided by the node manager 28 include transaction management, failure recovery, client/server request handling, runtime administration, etc. (*Id. at Col 2, lines 12-17*). All nodes 12 may be controlled globally by an administrator from a workstation 18 ("global" mode) or may singularly be controlled a node-by-node basis ("single node" mode). (*Id. at Col 3, lines 37-44*). This is accomplished via a transaction processing system management (TPSM) utility 32.

The TPSM utility 32 is a user interface and object-based data representation of the system 10, that allows a user to specify system 10, nodes 12, and/or modular components 20 that are to be manipulated by the systems management tools that provide management support for diagnostics, connection and disconnection from the system 10, and other system management functions. *(Id., Col. 4, lines 18-24; lines 62-64)*. The TPSM utility 32 comprises one or more object programs that are linked to one or more component libraries known as an SM Applications Programming Interface ("SM API"), and which provide the functions necessary for the desired administrative functions discussed above. All objects form a hierarchy that is representative of the physical hierarchy (i.e., session→system→node→component), and each object describes the status of the element that it represents (e.g., node object 50 describes the status of a particular node 12 of system 10). As is common in all object-based systems, each object has attributes, services and relationships with other objects. *(Id., Col. 5, lines 5-51)*.

It is clear from the discussion above that Fanshier does <u>not</u> teach or suggest Applicant's invention. First, although Fanshier discloses a distributed application environment (such as OLTP), there is no mention of a distributed network application <u>that requires centralized administration via a master node</u> (such as BEA System's Tuxedo® transaction manager and messaging middleware). This is a critical element in Applicant's invention, as it is the problem of running such an application in a clustered computing environment (such as MSCS) - where administration is controlled by a resource DLL (or the like) on the node to which the operation is directed (i.e., locally) - that Applicant is addressing. In fact, Fanshier clearly <u>teaches away</u> from the invention of Applicant's invention, since each node 12 has its own node manager 28 *(see, e.g., Fig. 1)*.

The Examiner's indication that the systems 10, nodes 12 and/or components 20 of the administrative request are equivalent to Applicant's "master node" is incorrect. Again, as set out above and more detail in the specification of the instant application, a "master node" performs administration for the <u>entire</u> distributed application. None of system 10, nodes 12 and/or components 20 can be considered "master nodes," as none controls administration for all of the nodes. The Examiner also erroneously indicates that Fanshier's "*coordinate[d] processing among nodes 12*" is equivalent to a distributed network application that requires centralized administration is in error. Mere *coordination* among components or nodes is not equivalent to a centralized *control* or administration via one master node.

Furthermore, in looking to Applicant's invention of claim 1, Fanshier does not teach the step of determining whether the originating node is a designated master node for the distributed network application. Examiner states that Fanshier teaches this at Block 122 where the ADMIN process 40 translates the command information into an appropriate message, and then locates the targeted element (system 10, node 12, component 20). There is no "determination" at step 122, just a response to the command that affects the system, node, and/or component. Applicant believes that perhaps the block to which Examiner meant to refer was Block 120 (misidentified as Block 100 in Fig. 6). However, the determination at the point is merely that of determining whether an input is a command from the TPSM utility 32. (*Fanshier, Col.15, lines 6-8*). It is clear that this is in no way equivalent to Applicant's invention where a determination is made as to "*whether the originating node is a designated master node for the distributed network application.*"

Fanshier also does not disclose the step of "*routing the administrative request from the originating node to the designated master node if the originating node is not the designated master node.*" The Examiner states that the ADMIN process' forwarding of administrative requests to the desired TOP END™ nodes 12 and modular components 20 is an equivalent. According to Fanshier, the TPSM utility 32 may be run in either local or remote mode. (*Fanshier, Col.4, lines 45-46*). In local mode, the TPSM utility 32 makes requests directly to the ADMIN process 40, which then forwards the administrative requests to the desired node or component and returns the responses to the node. In remote mode, the TPSM utility 32 makes this request through a transport process 42, which then forwards the request to an ADMIN process 40 on the appropriate node. According to Fanshier, the ADMIN process 40 on that node performs the desired function, transmits it to another modular component 20 on that node 12 or another node 12, and forwards responses back to the first node 12. (Applicant would note that it is note clear why the ADMIN process 40 actually performs the function and not the targeted node and/or component). While Applicant would agree that the requests in both cases of Fanshier's system could broadly be considered requests from "originating nodes," there is again no teaching of routing the request to the designated master node if the originating node is not the designated master node. Routing is not dependent upon master node determination, but simply a case of routing the request to the appropriate node and/or component that the administrator wishes to affect.

For the foregoing reasons, it is respectfully submitted that independent claim 1 is patentable over Fanshier. Independent claims 5 and 8 are substantially similar to claim 1, and the reasons for patentability as applied to claim 1 apply to claims 5 and 8 as well. As claim 9 depends directly from claim 8, it is submitted that it, too, is patentable over Fanshier.

Claims 2-4, 6, 7, 10 and 11 are rejected as being unpatentable over Fanshier in view Bendert. Bendert teaches a system of off-loading server operations without the partitioning of the object affected by the server operation and without off-loading the entire affected object. The Examiner states that Bendert teaches facilitating communication in a distributed processing system through the use of named pipes. However, Bendert also does not teach or suggest either alone, or in combination with Fanshier, Applicant's invention of enabling a distributed network application that requires centralized administration via a master node to execute on the nodes of a clustered computing environment, by receiving an administrative request from the clustered computing environment at an originating node, determining whether the originating node is a designated master node for the distributed network application, and then routing the request to the master node if it is determined that the originating mode is not the designated master node.

Thus, for the foregoing reasons, Applicant respectfully submits that independent claims 1, 5, and 8, and dependent claim 9 are patentable under 35 U.S.C. §102(e) over Fanshier. Furthermore, dependent claims 2-4, 6, 7, 10 and 11 are patentable under 35 U.S.C. §103(a) over Fanshier in view of Bendert. Reconsideration of these claims and the application as a whole is thus respectfully solicited.

It is believed that all of the stated grounds of rejection have been properly traversed, accommodated, or rendered moot. It is further believed that a full and complete response has been made to the outstanding Office Action, and as such, the present application, including claims 1-11, is in condition for allowance. Applicants therefore respectfully request prompt and favorable consideration of this amendment. If the Examiner believes that personal communication will expedite prosecution of this application, the Examiner is invited to telephone the undersigned at (215) 986-5169.

Respectfully submitted,

Lise A. Rode
Attorney for Applicant
Reg. No. 37,226

Unisys Corporation
Unisys Way, M/S E8-114
Blue Bell, Pennsylvania 19424-0001

The Director for Patents is hereby authorized to charge payment to Deposit Account No. 19-3790 of any fees associated with this communication.

I hereby certify that this correspondence is being transmitted via facsimile ((703) 746-7239) to the United States Patent and Trademark Office on the date shown below.

June 2, 2003

Michele S. McDevitt